

# Computing & Computer Science Department Curriculum Map

A coherent, spiral curriculum from Year 7 to Year 13 — developing digitally literate, technically capable and ethically informed young men who can understand, design and build the systems that shape modern life.

## 1 INTENTION

### Our Curriculum Vision

Why we teach Computing & Computer Science — the aims shaping every lesson from Year 7 to Year 13.

#### COMPUTATIONAL THINKERS

Pupils who decompose problems, recognise patterns and express solutions as precise, logical algorithms. From Scratch in Year 7 to Python sub-programs, trace tables and the OCR Exam Reference Language in Year 11, every year sharpens the ability to think algorithmically and reason about abstract systems.

#### CONFIDENT PROGRAMMERS

Pupils who move from block-based programming in Year 7 through Python fundamentals in Year 8, GCSE programming constructs across Years 9–11, to professional-standard application development in Years 12–13 — building genuine, transferable coding fluency, not surface familiarity.

#### ETHICAL DIGITAL CITIZENS

Pupils who understand the legal, ethical, cultural and environmental consequences of computing — the Computer Misuse Act and data protection from Year 7, to privacy, the digital divide and stakeholder analysis at GCSE, to professional security, testing and software lifecycles at Level 3.

ALGORITHMS

PROGRAMMING

DATA & SYSTEMS

NETWORKS

SECURITY

BOOLEAN LOGIC

ETHICS & LAW

APP DEVELOPMENT

### A Spiral Curriculum

Seven golden threads — algorithms, programming, data representation, networks, security, ethics/law and application design — are introduced in Year 7 at their simplest and grow in complexity every year. Binary representation first met in Year 7 hardware deepens into hexadecimal, image and sound representation at Year 9 GCSE depth, and underpins IP addressing and compression in Year 10. Cyber security introduced in Year 7 becomes Boolean logic gates in Year 8, network security protocols in Year 10, and defensive programming, validation and authentication in Year 11. Block-based Scratch in Year 7 transitions to Python in Year 8, to the full GCSE programming toolkit across Years 9–11, and to professional application and website development at Level 3. PATHS feedback is a consistent feature of every year from 7 to 13, making every assessment a genuine learning event.

## 2 IMPLEMENTATION

THE COMPUTING JOURNEY — YEARS 7 TO 13. Each unit is sequenced to build on prior knowledge. Seven golden threads spiral across all seven years alongside an ambitious body of substantive knowledge and growing programming fluency.

| KEY STAGE 3 — COMPUTING · KNOWLEDGE-RICH, SPIRAL PROGRAMME BUILDING FROM BINARY AND BLOCK PROGRAMMING TO PYTHON AND GCSE FOUNDATIONS · ONE LESSON PER FORTNIGHT · FIVE TOPICS PER YEAR |                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7<br>KS3                                                                                                                                                                               | <b>Understanding Computers · Cyber Security</b><br>Hardware vs software, the CPU, binary representation, data storage, baseline assessment. Cyber security follows: malware, hacking exploits, the Computer Misuse Act, data protection, copyright and ethical responsibilities.                      | <b>Digital Graphics · AI and Machine Learning</b><br>Binary representation of images — vector vs bitmap, resolution, colour depth — with practical Photoshop work. AI and ML: rules, training data, pattern recognition, bias and ethics in algorithmic systems.                                      |
| 8<br>KS3                                                                                                                                                                               | <b>Cyber Security · Computational Thinking &amp; Logic</b><br>Cyber threats deepened: phishing analysis, dictionary and brute-force attacks, copyright law. Boolean logic: AND, OR and NOT gates, truth tables, compression and abstraction — the conceptual bridge from hardware into programming.   | <b>Networks · Introduction to Python</b><br>Networks: packets, bandwidth, encryption and topologies. Python transition from Scratch: variables, data types, selection and iteration — sequenced after Boolean logic is secure so conditions in code are conceptually grounded.                        |
| 9<br>KS3 / GCSE FOUNDATION                                                                                                                                                             | <b>Systems Architecture (SLR 1.1) · Python Fundamentals</b><br>Theory: fetch-execute cycle, Von Neumann architecture, CPU performance and embedded systems. Programming in parallel: variables, data types, selection, pseudocode design, iteration — Learning Portfolio assessment at end of Autumn. | <b>Memory &amp; Storage (SLR 1.2) · Python Continued</b><br>Theory: RAM vs ROM, virtual memory, secondary storage, data units, binary/hexadecimal, character sets, image and sound representation, compression. Python: constructs consolidated; Time2Code independent practice running continuously. |

| KEY STAGE 4 — OCR GCSE COMPUTER SCIENCE (J277) · SIX TOPICS ACROSS YEARS 10–11 · PAPERS J277/01 (COMPUTER SYSTEMS) AND J277/02 (COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING) SAT IN SUMMER YEAR 11                                                                                                        |                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>GCSE Y1                                                                                                                                                                                                                                                                                                    | <b>Networks &amp; Protocols (1.3) · Network Security (1.4)</b><br>LAN/WAN, topologies, DNS, protocols and layering; a weekly independent Python strand throughout. Network Security: malware, phishing, DoS, SQL injection, penetration testing, firewalls and encryption. System Security game for consolidation. | <b>Systems Software (1.5) · Ethical, Legal &amp; Environmental (1.6)</b><br>OS functions, utility software — encryption, defragmentation, compression. Ethical/Legal/Cultural/Environmental: DPA 2018, Computer Misuse Act, CDDPA 1988, open-source vs proprietary, digital divide and the Telium extended project. |
| 11<br>GCSE Y2                                                                                                                                                                                                                                                                                                    | <b>Producing Robust Programs (2.3) · Paper 2 Revision</b><br>Defensive design: input validation, authentication, maintainability. Systematic testing: normal, boundary, invalid and erroneous data; iterative vs final testing; syntax vs logic errors. Dedicated Paper 2 revision interleaved throughout.         | <b>Boolean Logic (2.4) · Languages &amp; IDEs (2.5)</b><br>AND, OR and NOT gates; one- and two-level logic diagrams and truth tables. High- and low-level languages; Little Man Computer assembly; compilers, interpreters, assemblers; IDE facilities. Full mock examinations with PATHS feedback.                 |
| <b>Programming Fundamentals (2.2) · Algorithms (2.1)</b><br>'Code first, formalise second': full Python toolkit — constructs, data types, file handling, SQL, arrays and sub-programs. Algorithms then formalises: abstraction, decomposition, searching, sorting, trace tables and OCR Exam Reference Language. |                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                     |
| <b>Full GCSE Revision — Both Papers</b><br>Systematic past-paper practice across J277/01 and J277/02; Smart Revise spaced retrieval across every SLR; targeted PATHS intervention on mock-analysis weakest areas; walking-talking mocks and final Python programming practice.                                   |                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                     |

| KEY STAGE 5 — OCR CAMBRIDGE ADVANCED NATIONAL IN COMPUTING: APPLICATION DEVELOPMENT (LEVEL 3 AAQ) · F160 + F164 IN YEAR 12 · F161 + F162 + F166 IN YEAR 13 · EXAMINATIONS + NEA PORTFOLIOS                                                                                                         |                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12<br>AAQ Y1                                                                                                                                                                                                                                                                                       | <b>F160 — Fundamentals of Application Development (Exam, ≈90 GLH)</b><br>Software development models, systems analysis, client requirements, HCI and interface design principles, project planning — taught Autumn 1 through Spring 2; exam revision in Spring 2. F160 theory informs and improves F164 NEA quality.           | <b>F164 — Website Development (NEA, ≈56 GLH)</b><br>NEA introduced Autumn 2: website brief analysis, planning, initial design documentation. Technical development — HTML, CSS, JavaScript, responsive design — runs Autumn 2 through Summer 2. Portfolio evidence gathered continuously alongside F160.                   |
| 13<br>AAQ Y2                                                                                                                                                                                                                                                                                       | <b>F161 — Developing Application Software (Exam) · F162 NEA Tasks 1–2</b><br>F161 taught Autumn 1–2: application platforms, storage, data formats, APIs, protocols, security, testing and legal considerations. Mock exam in Autumn 2; F161 external examination in January. F162 UX/UI NEA runs simultaneously from Autumn 1. | <b>F162 NEA Completion · F166 — Software Development (NEA) Begins</b><br>F162 hi-fi prototype and showcase/communication evidence completed post-January; moderated Summer 1. F166 teaching begins Spring 2: design principles, System Design Specification and independent coding. All three units simultaneously active. |
| <b>F160 Exam + F164 Final Submission</b><br>F160 external written examination (OCR, May). After the exam, all taught time supports completion, refinement and final submission of the F164 NEA portfolio by Summer 2. PATHS-structured feedback throughout both units with DIRT time.              |                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                            |
| <b>F166 Development, Testing &amp; Final Submission</b><br>F166 dominates Summer 1–2: technical development, systematic testing evidence, critical evaluation and final portfolio completion. F166 moderated Summer 2. Full qualification complete: F161 examination + F162 + F166 NEA portfolios. |                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                            |

## 3 IMPACT

### How We Measure Success

What our pupils know, can do, and go on to achieve.

#### ACADEMIC OUTCOMES

- Strong OCR GCSE outcomes (J277) with high attainment at grades 7–9.
- Year-on-year improvements in PATHS assessment data demonstrating genuine pupil progress.
- Level 3 AAQ results meeting or exceeding national averages at Merit and Distinction.
- High-quality F162 and F166 NEA portfolios demonstrating genuine independent application development.
- Successful progression of Year 11 pupils into the Level 3 AAQ computing pathway.

#### WIDER IMPACT

- Every pupil leaves Year 11 able to write, test and debug Python programs independently.
- Computational thinking — decomposition, abstraction, algorithmic reasoning — transferable across every academic subject.
- Digital literacy and ethical awareness: pupils understand the legal and societal consequences of computing decisions.
- Cultural capital: the history and folklore of computing, the great debates of the digital age, careers across the fastest-growing economic sector.
- Computing chosen as a post-16 pathway into technology, engineering, mathematics and science degrees and apprenticeships.

#### KNOWLEDGE

From binary and the CPU to GCSE systems architecture, protocols and ethical/legal frameworks — built, revisited and deepened across seven years.

#### SKILLS

Python programming from Scratch to professional application development; algorithm design, testing and debugging; NEA portfolio production.

#### DISPOSITION

Resilience through debugging; intellectual curiosity through 'wonder questions'; integrity through honest testing; responsibility as makers, not just consumers.

#### DESTINATIONS

University CS, mathematics and engineering; software development, cyber security, data science, AI, network engineering and technology entrepreneurship.